REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1993	Final Report - 9/30/89 to 3/31/93

4. TITLE AND SUBTITLE

High-Performance Control of Multi-Link Flexible Articulated Space Structures

5. FUNDING NUMBERS

AFOSR-89-055-A-C

89-0555

6. AUTHOR(S)

Professor Robert H. Cannon, Jr.
Professor Stephen M. Rock
Bill Ballhaus
Ed Wilson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Stanford University Aerospace Robotics Laboratory
Department of Aeronautics and Astronautics
Stanford, California 94305

8. PERFORMING ORGANIZATION REPORT NUMBER

AFOSR-TR- 93 0665

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. James Chang
AFOSR/NA
Building 410
Bolling AFB 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release,
distribution unlimited

12b. DISTRIBUTION CODE

93-20244

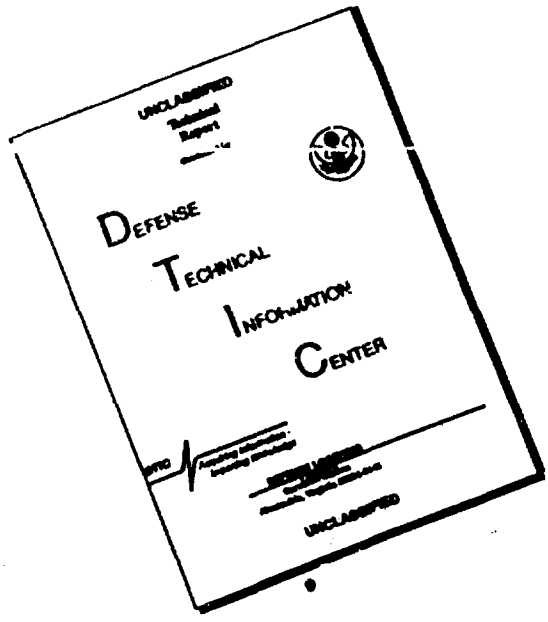
13. ABSTRACT (Maximum 200 words)

This is the final report for contract AFOSR-89-055-A-C. This contract consisted of an original three year contract and a subsequent six month extension. In this research program, five specific tasks were identified. Research corresponding to the first three tasks has been reported previously, and is not presented in detail in this document. A detailed description of the work performed under Tasks IV and V is given in this final report. Specifically, this document discusses research performed in: (1) developing a total system approach to controlling flexible structures based on distributed sensing/actuation, and (2) developing a hybrid neural network approach for flexible structure control.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	24
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- DOE - See authorities.
- NASA - See Handbook NHB 2200.2.
- NTIS - Leave blank.

Block 12b. Distribution Code.

- DOD - Leave blank.
- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- NASA - Leave blank.
- NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

High-Performance Control of Multi-Link Flexible Articulated Space Structures

AFOSR Final Report for Contract AFOSR-89-055-A-C

Stanford University Aerospace Robotics Laboratory
Department of Aeronautics and Astronautics
Stanford, California 94305

Principal Investigators:
Professor Robert H. Cannon Jr.
Professor Stephen M. Rock

July 1993

AFOSR Final Report for Contract AFOSR-89-055-A-C

Stanford University Aerospace Robotics Laboratory
Stanford, California 94305

June 1993

Introduction

This is the final report for contract AFOSR-89-055-A-C. This contract consisted of an original three year contract and a subsequent six month extension. In this research program, five specific Tasks were identified. They are:

- Task I: Advance the fundamental science of modelling nonlinear, multi-link, flexible articulated structures.
- Task II: Extend the powerful theory of automatic control by specific increments, enabling (*for the first time*) end-point control of nonlinear, multi-link, flexible, articulated structures.
- Task III: Build the new theoretical foundation for achieving adaptive control of nonlinear, multi-link articulated, redundant structures whose physical parameters are unknown.
- Task IV: Pursue a total system approach to controlling flexible structures based on distributed sensing/actuation.
- Task V: Develop a hybrid neural network approach for flexible structure control.

Research corresponding to the first three tasks has been reported previously, and will not be reported in detail here. Research in Tasks IV and V are presented in this final report.

Summarizing briefly, the results of all five tasks were:

- Task I Findings: Celia Oakley broke new ground in this arena when she developed a new modelling technique, the System Modes Representation, that makes it possible to represent the dynamics of complex dynamic structures very accurately with *low order models*. Celia's thesis, which fully documents this research, was submitted previously to the AFOSR.
- Task II Findings: Celia was able to demonstrate, for the first time anywhere, experimental end-point control of a a two-link, extremely flexible structure. Building on her work, Bill Ballhaus developed and experimentally verified a new fundamental control strategy for quickly and accurately repositioning lightly-damped, multi-link, flexible space structures. His new theory, based on time-varying controller gains, not only repositions the structure in "near minimum-time", but also generates the optimal trajectory for the repositioning. A report on Bill's work, along with a publication entitled "Optimal Control of a Two-Link Flexible Manipulator Using Time-Varying Controller Gains: Initial Experiments", was submitted in October 1991 in an AFOSR annual report.

- Task III Findings: Vincent Chen developed a new general adaptive control framework - the adaptive task-space framework - that is capable of providing full adaptation to large systems, including, for example, multiple robots with any number of manipulators. Vince's work was fully reported in the October 1991 AFOSR annual report, and his thesis was sumbitted to the AFOSR on March 25, 1993.

- Task IV Findings: Bill Ballhaus has added very quick tip actuators at the end of a multi-link flexible structure in an attempt to reduce dramatically the bandwidth limitations imposed by inherent structural flexibility. Initial research demonstrated the advantages of tip actuation in terms of increased end-point performance. In addition, coupling issues associated with the tip actuation were identified. Research focused on addressing these coupling issues resulted in the development of a new, extended end-point controller. This research has been, and will continue to be, heavily drawn from in developing the modular control framework for multi-link flexible structures under our current research contract. A detailed description of this work is given in Section I of this document.

- Task V Findings: Ed Wilson worked towards understanding the applicability of neural networks for control of flexible structures. The underlying philosophy of this research was to blend neural network design techniques with proven control theory. This work began with applying neural networks to a non-linear rigid-body problem. A publication generated from this research, "Experiments in Control of a Free-Flying Space Robot Using Fully-Connected Neural Networks", is included in the Appendix of this document. Furthermore, very initial investigations and experiments were performed in the control of a flexible beam using neural networks. A detailed description of this research is given in Section II of this document.

This report consists of two main sections and an appendix. Section I gives a full report on Task IV, Section II reports on Task V, and the Appendix includes a paper written during this reporting period on the research performed in Task V.

Section I: Total System Approach to Controlling Flexible Structures

1 Introduction

Inherently flexible structures with discrete actuation are limited in the speed and precision at which they can be repositioned. In order to remove these fundamental limits, the ARL is developing a fundamentally new approach to controlling these systems based on a total system perspective which integrates distributed actuation and sensing.

As a first step toward distributed sensing and actuation, we proposed under Task I of the six month extension to investigate the use of local tip actuation to achieve high-performance end-point control of multi-link flexible structures. Specifically, we proposed to: 1) demonstrate experimentally the advantages and feasibility of local tip actuation, 2) identify and understand fundamental research issues which will facilitate development of a control approach based on distributed actuation and sensing.

A report of this research is given below. The specific accomplishments include:

1. the development of an experimental test bed to investigate the use of local tip actuation,
2. initial results which demonstrated the advantages of local tip actuation,
3. the identification of fundamental dynamic coupling issues associated with local tip actuation, and
4. the development of an extended flexible structure end-point controller which compensates for the coupling effects introduced by the tip actuation.

2 Experimental Test Bed Development

In order to demonstrate experimentally the feasibility of local tip actuation, we added very quick and precise tip pointing mechanism to the end of our existing multi-link flexible structure. The resulting experimental test bed is shown in Figure 1. The tip mechanism, or mini-manipulator, consists of a five-link, closed-kinematic chain driven by two small electric DC motors. The entire system is supported by two air-bearing cushions,

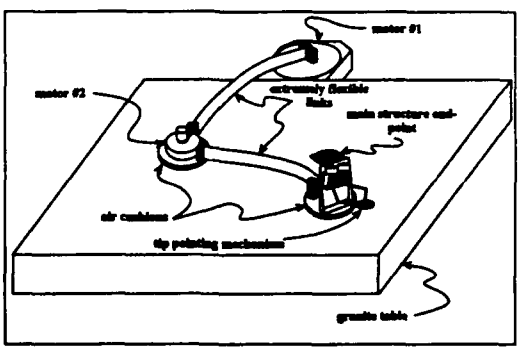


Figure 1: **Experimental Hardware Schematic**

Shown in this figure is the experimental multi-link flexible structure containing a very quick and precise tip pointing mechanism or mini-manipulator.

and operates in the horizontal plane. This non-linear, two-way dynamically coupled experimental apparatus is analagous to a precision pointing mechanism mounted at the end of a large, flexible space structure.

The notation that will be used to describe the experimental apparatus is given in Figure 2. X_{eb} and Y_{eb} describe the tip position of the main arm, θ_{eb} refers to the absolute rotation angle at the tip of the main arm, and X_{mini} and Y_{mini} indicate the absolute end-point position of the tip mechanism or mini-manipulator.

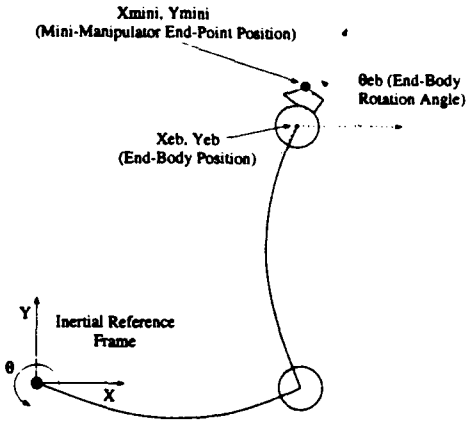


Figure 2: **Notation**

3 Initial Experimental Results

As specified in the six month extension, the goal of this research was to demonstrate rapid repositioning of the entire space structure, while at the same time maintaining precision pointing of the tip mechanism. Initial experiments [1] [2] demonstrated the benefits of local actuation. Using a partitioned control structure (one in which a system approach is NOT used – rather the two-way coupling is ignored and controllers for the structure and the tip mechanism are designed independently), the precision pointing mechanism is able to keep errors small even in the presence of relatively large vibrations of the space structure. The specific control structure for these initial results consists of independent joint PD controllers on the flexible structure and and end-point based controller on the mini-manipulator. Figures 3 and 4 illustrate the effectiveness of tip actuation in maintaining accurate tip positioning in the presence of large disturbances. The flexible

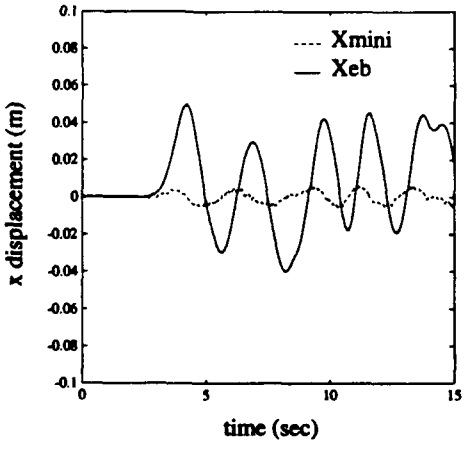


Figure 3: **End-Point Position Control: x displacement.**

Small displacement of the tip pointing mechanism despite a large forced motion of the tip of the main structure.

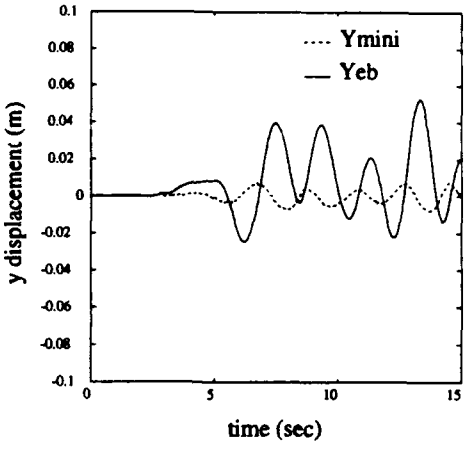


Figure 4: **End-Point Position Control: y displacement.**

Small displacement of the tip pointing mechanism despite a large forced motion of the tip of the main structure.

structure is excited by an externally applied force. The corresponding displacement of the controlled tip from its desired position is kept very small by the tip pointing mechanism, even in the presence of large motions of the main structure end-point.

Figures 5 and 6 show a coordinated repositioning of the flexible structure. These position responses reiterate the advantages of speed and precision offered by the local actuation: once the main structure brings the tip actuator within range of the target (desired tip position), the tip pointing mechanism is able to achieve small end-point errors long before the main structure comes to rest.

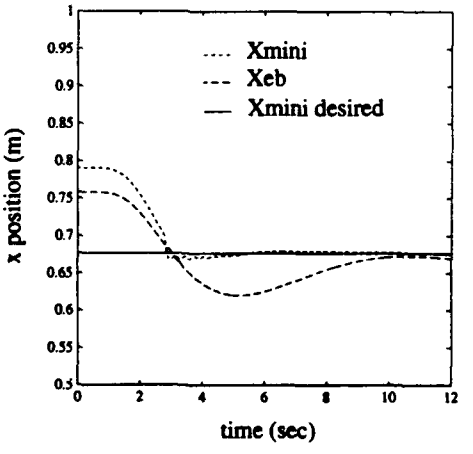


Figure 5: **Coordinated Repositioning: x motion.**

Position responses of the tip pointing mechanism and of the tip of the main flexible structure during a repositioning maneuver.

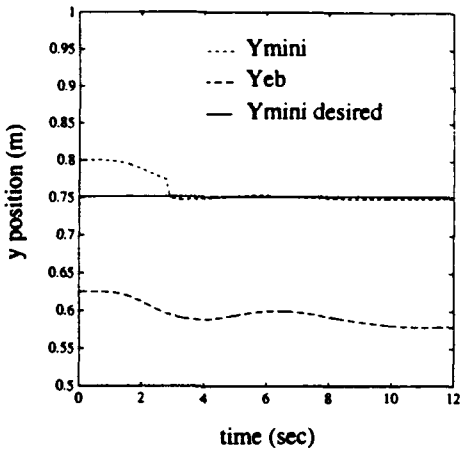


Figure 6: **Coordinated Repositioning: y motion.**

Position responses of the tip pointing mechanism and of the tip of the main flexible structure during a repositioning maneuver.

4 Fundamental Dynamic Coupling Issues

From these initial experimental results, significant dynamic coupling interaction between the tip actuator and the flexible structure was discovered [3]. The coupling effects are most noticeable in the plot of the rotation angle of the main arm end-point shown in Figure 7. At approximately five seconds, the main arm has positioned the mini-manipulator such that the desired end-point position is within reach. As the end-point controller attempts to zero the end-point error, reaction forces are imparted by the mini-manipulator on the main arm. These reactions induce a 3 Hz oscillation in the main arm, and result in rotation of the main arm end-body. For increased end-point controller gains, the coupling becomes more significant.

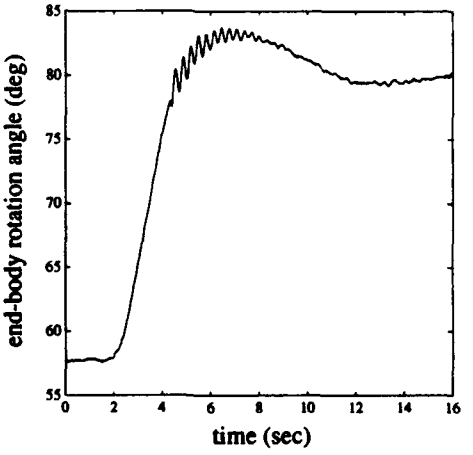


Figure 7: **Tip Rotation of the Main Arm**

The tip rotation angle of the main arm is displayed in this figure. Dynamic coupling between the mini-manipulator and the main arm is evident even for a low gain mini-manipulator end-point controller.

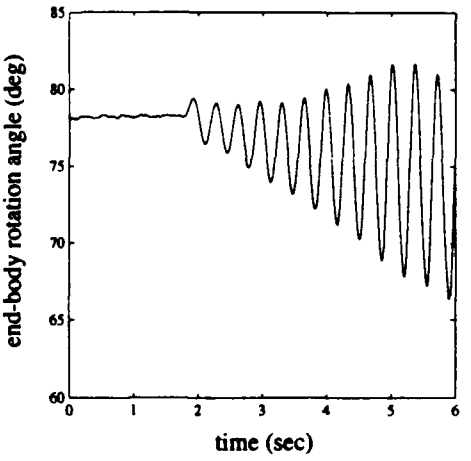


Figure 8: **Tip Rotation of the Main Arm**

Unmodelled coupling effects can lead to instability. This figure demonstrates this instability for the case where the main arm PD controller gains are zero.

Increasing the gains further can result in instability. As the stabilizing effects of the main arm PD controller become negligible compared to the coupling interaction caused by the mini-manipulator end-point controller, the system becomes unstable. Figure 8 demonstrates this instability for the case where the PD controller gains are zero.

Figure 9 shows the collocated transfer functions of the main arm with the mini-manipulator mounted on its tip. The first system mode, shown in the elbow transfer function plot, occurs at approximately 3 Hz. Because of the relatively large tip and hub masses, the second link closely resembles a beam with pinned-pinned boundary conditions. As a result, the tip motion corresponding to the first system mode is primarily a rotation of the tip of the second link. The response shown in Figure 8 indicates that the unmodelled coupling effects destabilize the first system mode.

5 Extended Flexible Structure End-Point Control

It has been well known since Eric Schmitz's work performed at the ARL [4] that in order to achieve high-performance control of flexible structures, end-point sensing must be used. This idea was then extended to multi-link manipulators by Celia Oakley at the ARL [5].

The current research has demonstrated that end-point position information, although necessary, may not be sufficient to achieve high-performance control. To explain this idea, consider the first system mode shape of the flexible two-link structure with the mini-manipulator shown in Figure 10. The first system mode closely resembles a beam with pinned-pinned boundary conditions. As a result, if only the position or lateral deflection of the end-point is sensed, it will be difficult to detect excitation of this mode. Similarly, with respect to control design, achieving good control of the end-point lateral deflection does not insure good control of the first system mode. Thus, by studying the system mode shapes, it can be concluded that in order to achieve high-performance control, tip rotation, as well as position, must be sensed. Similarly, the controller should be designed to control the end-point rotation as well as position.

The end-point control strategy developed by Oakley was extended to control tip rotation as well as position. Figure 11 shows the closed loop roots when the controller is only trying to control lateral deflection of the flexible structure. Figure 12 demonstrates the effect of controlling tip rotation as well as lateral deflection.

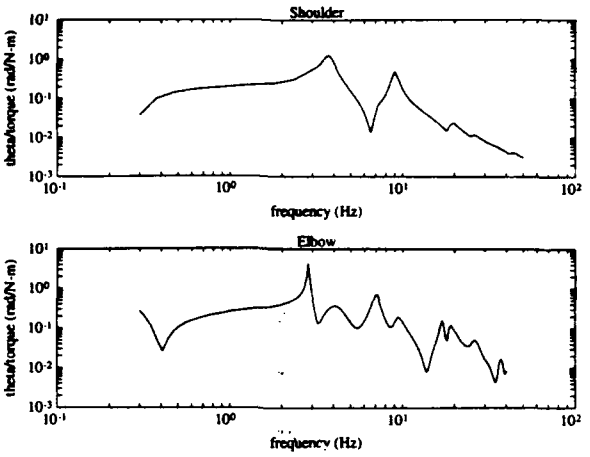


Figure 9: **Experimental Frequency Response**

This figure shows the experimental elbow joint collocated transfer functions for the main arm with the mini-manipulator mounted at its tip.

First Theoretical Flexible Mode Shape

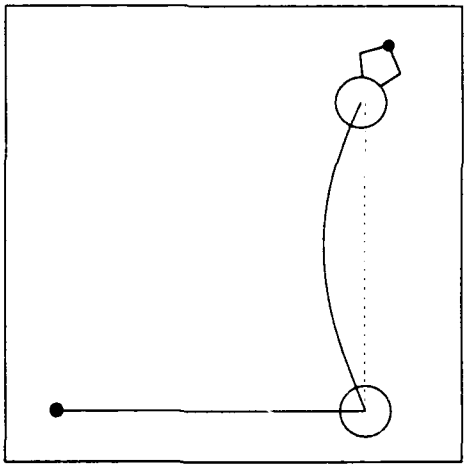


Figure 10: **First Theoretical System Mode Shape**

Namely, greater damping of the first and third system modes can be achieved.

The performance of the extended end-point controller is illustrated in Figures 13, 14, and 15, and in the accompanying video. Figure 13 and 14 illustrate a coordinated slew of the flexible system using the end-point controller described in the last section. Figure 15 shows the corresponding tip rotation during the slew. A comparison of Figure 15 with Figure 7 indicates that this end-point controller does a significantly better job of controlling the first system mode than the partitioned PD controller.

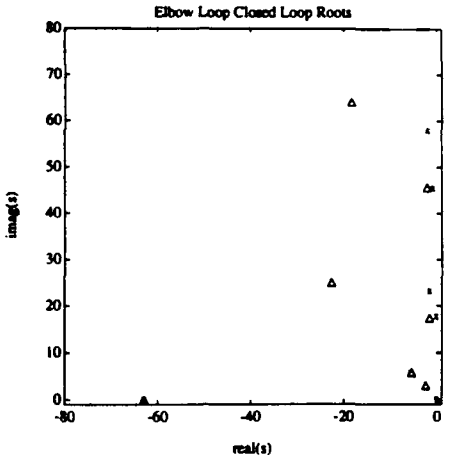


Figure 11: **End-Point Lateral Deflection Controller**

The triangles represent closed loop roots while the squares indicate open loop root locations. From this figure it is evident that controlling only the main arm end-point lateral deflection does not sufficiently damp the first and third system modes.

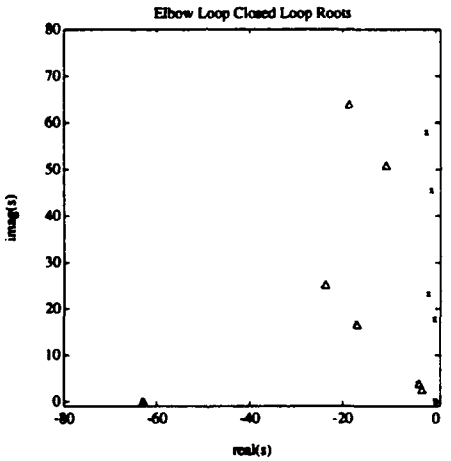


Figure 12: **End-Point Lateral Deflection and Tip Rotation Controller**

Controlling tip rotation as well as end-point lateral deflection provides significant damping of the first and third system modes.

6 Conclusions

This new research contributes to the fundamental understanding of multi-link, extremely flexible structures. We have demonstrated the advantages of local actuation in improving flexible structure end-point performance. The findings of this research will aid in the development of our system-integrated finite-element control approach.

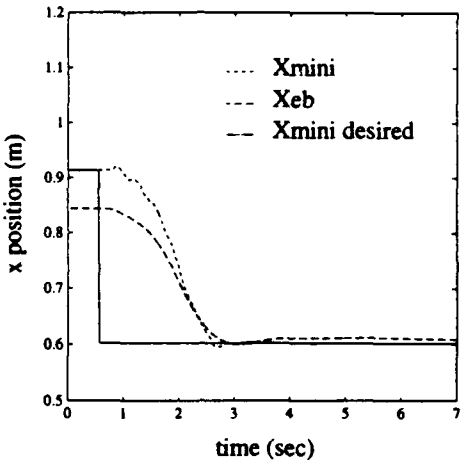


Figure 13: **Coordinated Repositioning: x motion.**

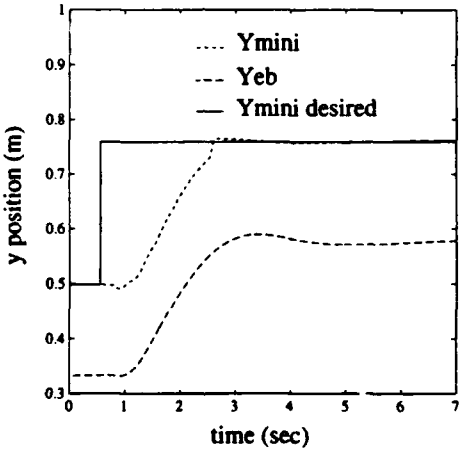


Figure 14: **Coordinated Repositioning: y motion.**

References

[1] W. L. Ballhaus and S. M. Rock. End-point control of a two-link flexible robotic manipulator with a mini-manipulator: Initial experiments. In *Proceedings of the American Control Conference*, Chicago, IL, June 1992.

[2] W. L. Ballhaus, L. J. Alder, V. W. Chen, W. C. Dickson, and M. A. Ullman. Stanford aerospace robotics laboratory research overview. In *Proceedings of the Space Operations, Applications and Research Symposium*, Houston, TX, August 1992.

[3] W. L. Ballhaus and S. M. Rock. End-point control of a two-link flexible robotic manipulator with a mini-manipulator: Dynamic coupling issues. In *Proceedings of the ASME Winter Annual Meeting*, Anaheim,

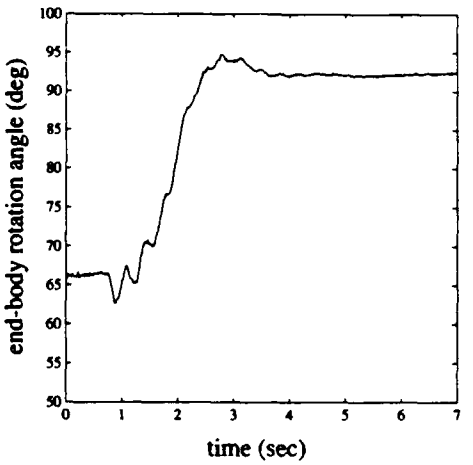


Figure 15: **Tip Rotation of the Main Arm**

Extending the end-point position controller to control the tip rotation as well provides significant damping of the first flexible mode.

CA, November 1992.

[4] Eric Schmitz. *Experiments on the End-Point Position Control of a Very Flexible One-Link Manipulator.* PhD thesis, Stanford University, Department of Aeronautics and Astronautics, Stanford, CA 94305, June 1985. Also published as SUDAAR 548.

[5] Celia M. Oakley. *Experiments in Modelling and End-Point Control of Two-Link Flexible Manipulators.* PhD thesis, Stanford University, Department of Mechanical Engineering, Stanford, CA 94305, April 1991.

Section II: Development of Hybrid Neural Networks for Control

1 Introduction

Neural networks, due to their capacity for adaptation, learning, and very high computational throughput, have the potential to cause profound fundamental advances in control. In some cases their learning abilities and inherent non-linear nature allow them to solve control problems and provide performance unmatched by conventional methods. In other cases their distributed nature and resulting computational power allow them to implement known solutions that were previously impossible to implement in real-time.

2 Neural Networks for Control

Work was done towards understanding the applicability of neural networks for control of flexible structures. A large part of this work involved basic control work on a rigid-body problem - specifically the base control problem for our free-flying space robots. Unexpectedly, the non-linear aspect of this problem presented by the on-off air thrusters proved to be sufficiently rich in research issues that this area has become the focus of our current work in neural network control. This work is reported in the Appendix.

Reported in this section is some very preliminary work on the application of neural networks for control of flexible structures. Initial investigations and some experiments have been performed on a flexible beam with a single actuator. Further research in this area is very promising, but is beyond the scope of current funding.

2.1 Fully-Connected Neural Network Architecture

One research philosophy we have tried to maintain is to blend neural network design techniques with proven control theory. One example is outlined in the attached paper, "Experiments in Control of a Free-Flying Space Robot Using Fully-Connected Neural Networks" which will be presented at the World Congress on Neural Networks in Portland, Oregon in July 1993 [1]. This paper outlines an expanded architecture for traditional feed-forward neural networks, which was partly inspired by basic control concepts. That concept is that although truly linear physical systems do not exist, many systems can be closely approximated by a linear model. A linear controller is often capable of achieving stabilization, and a reasonable level of performance. The non-linear and adaptive benefits of the neural network will more easily be applied if introduced to augment a working controller rather than build one from scratch.

Our architecture provides for the seamless direct input of a known linear solution, or the very quick adaptation of a linear solution. Once a certain level of performance has been obtained with the linear portion of the network, the remaining non-linear segment is adapted to give the performance increase that non-linear control can often provide. An additional benefit is that by comparing the connection strengths of the linear and non-linear components, the degree of non-linearity can be assessed. If the solution is primarily a linear one, it can easily be extracted and analyzed using standard control analysis methods such as frequency response and root locus plots.

The WCNN paper presents an application of this architecture to the thruster mapping problem presented by the on-off thrusters on the free-flying space robot. By initializing the network with the easily calculable linear solution, learning performance is greatly accelerated. The extra connections provided by the architecture result in a more accurate thruster mapping. A complexity control technique is successfully implemented to handle the over-fitting problem that is introduced by the enhanced connectivity [2].

2.2 Initial Experiments on Single-Link Flexible Arm

Our understanding of basic control theory was again useful in our initial experiments on the single-link flexible arm. In these experiments, we developed collocated controllers using the powerful neural network training technique known as backpropagation through time [2] [3]. With this method, a neural network is run in

simulation mode with a model of the system to be controlled. After each run (regulation from an initial state in this case), the weights are adjusted to reduce the user-specified cost function resulting from the run.

For a linear controller using full-state feedback, with a linear plant, and an LQR-style cost function, this method produces the exact LQR solution. Only one initial condition for the simulation is required, as long as each mode is sufficiently excited. One benefit of a neural network approach is that it can handle non-linear controllers or non-linear plants, usually with little added complexity.

Since we were designing a collocated controller, the full state was not available, and only the hub angle and rate were used for feedback. Although the learning algorithm permits non-linear controllers and plants, to simplify analysis and facilitate our interpretation of what the neural network learning was doing, we used a linear model with one rigid and three flexible modes, and a linear controller acting on hub position and rate (a collocated PD controller - high probability of success for our first attempt).

One interesting observation was that our very first attempts with neural network-trained controllers resulted in controllers that performed very well on the simulated plant, but were unstable when tested on the real system. Analysis of root locus plots showed that the roots for the highest flexible mode had been moved to the verge of instability. In the simulation, this (almost) un-damped mode contributed little to the cost, allowing the gain to increase until just before it went unstable. Such a highly-tuned controller was easily unstabilized due to the errors in the model used for training. The over-tuning problem was made evident by the root locus plots, but its cause was not clear at first.

We found that adding non-zero initial conditions on the flexible modes solved the problem. The collocated controller was of sufficiently low bandwidth that with no initial conditions on the flexible modes, they were insufficiently excited, and the learning law did little more than to keep them marginally stable. To find a systematic way to choose excitations, we initialized each flexible mode with the same energy level. Bryson's rule, a procedure commonly used for LQR control design, was used to select the cost function that the neural network minimized.

Figure 1 below shows the response to initial conditions of the resulting controller. These initial conditions are the ones used during training, and only the amplitudes are shown here for clarity. Note that this is not a high-performance controller, but that is because we are fundamentally limited by only using hub information. Future experiments using endpoint information should provide significantly better performance. The root locus plot on the right shows the poles and zeroes of the torque-to-hub-angle transfer function. Unlike a traditional root locus plot where the roots move as a gain is varied, this one shows the closed loop system roots moving as the learning progresses.

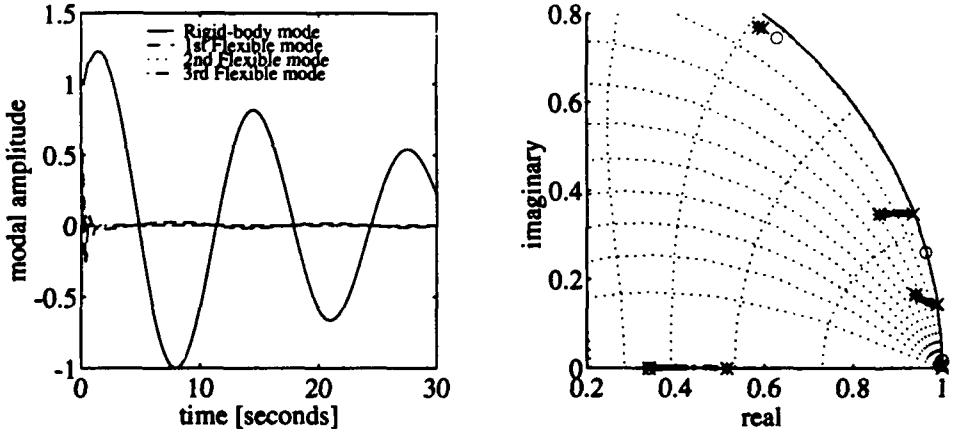


Figure 1: Collocated controller performance, training history

Figure 2 below compares simulation and experimental for the neural-network-designed collocated controller. The beam was given an initial displacement of 0.2 radians and commanded to return to zero. The mismatch indicates the modelling inaccuracy, as well as the robustness of the controller. Note that the robustness resulted from our application of basic control knowledge while developing the neural network. Our first attempt with the neural network was unstable.

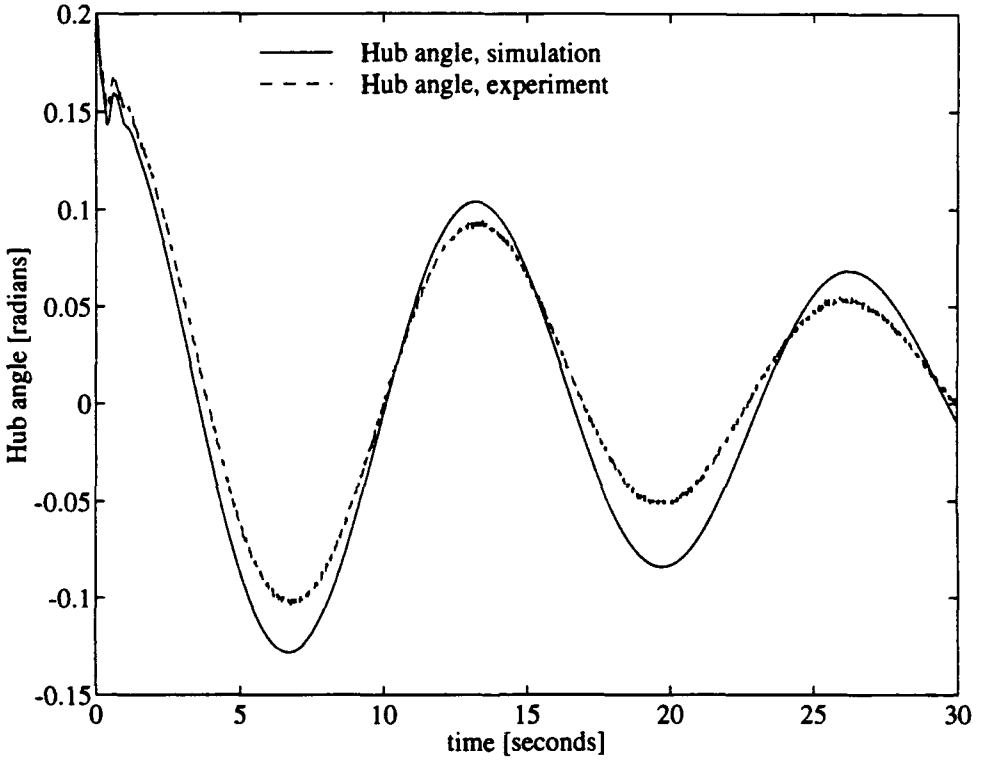


Figure 2: **Collocated controller performance, training history**

3 Conclusions and Future Work

Although future work on applying neural networks to the flexible systems appears promising, in the short term, our work on the free-flying space robot seems more fruitful, and our efforts will be concentrated there for now.

Our plans to draw on existing control theory for developing neural network controllers have worked well so far. The fully-connected architecture presented at the World Congress on Neural Networks allows direct integration of a linear solution. Understanding of basic control techniques such as root locus, LQR design, and system ID, have all directly or indirectly aided our development of our first controllers for the flexible-link manipulator.

References

[1] Edward Wilson and Stephen M. Rock. Experiments in control of a free-flying space robot using fully-

connected neural networks. In *Proceedings of the World Congress on Neural Networks*, Portland OR. July 1993.

[2] Andreas S. Weigend, Bernardo A. Huberman, and David E. Rumelhart. Predicting the future: A connectionist approach. *International Journal of Neural Systems*, 1(3):193–209, 1990.

[3] Paul J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, October 1990.

Appendix

Experiments in Control of a Free-Flying Space Robot Using Fully-Connected Neural Networks

Edward Wilson* Stephen M. Rock†
Stanford University
Aerospace Robotics Laboratory
Stanford, California 94305

Abstract

A general network architecture is presented that is capable of exploiting the full capabilities of the backpropagation algorithm. The extra connectivity of this architecture, which is unavailable in a layered network, allows seamless integration of linear *a priori* solutions, communication between output neurons, and greater overall functionality than a layered network. The increase in parameters can exacerbate over-fitting problems, and a systematic complexity control method is successfully demonstrated that lessens this problem. The viability of these methods is tested by applying them to a thruster mapping problem characteristic of space robots. Realizability is shown in an experimental demonstration on a 2-D laboratory model of a free-flying space robot.

1 Introduction

In the literature, the term "fully-connected feed-forward neural network", usually refers to a *layered* network, with an input layer, one or more hidden layers, and an output layer. "Feed-forward" indicates that signals flow from the input layer, through hidden layers, and to the output layer in one direction only, which is required by the backpropagation algorithm. "Fully-connected" indicates that every input is connected to every neuron in the first hidden layer, and so on between successive layers. While this layered architecture may be particularly well suited for many applications and certain hardware implementations, a more general structure may be able to take advantage of the full capabilities offered by the backpropagation algorithm [1].

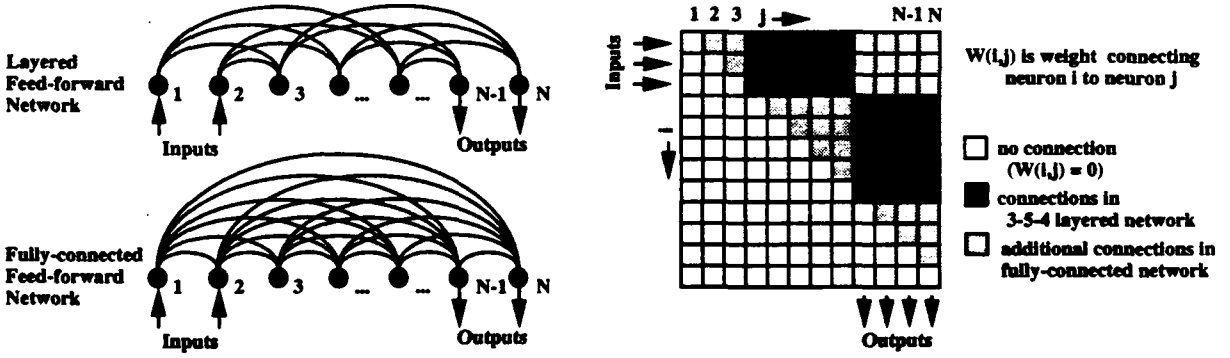


Figure 1: **Extra connections available with fully-connected feed-forward networks**

In this paper, the term "fully-connected" will refer to the structure shown in Figure 1. Instead of layers, a fully-connected network can be considered to have neurons that are ordered, beginning with the first input, ending with the last output, and having hidden units in between, perhaps interspersed among input or output units [2]. Backpropagation restricts information flow to one direction only, so to get maximum interconnections, each neuron takes inputs from all lower-numbered neurons and sends outputs to all higher-numbered neurons. For example, the last output neuron takes inputs from all the hidden neurons, just as in a layered architecture; however, it also takes inputs from each of the input neurons and previous output neurons.

*Ph.D. Candidate, Department of Mechanical Engineering. Research partially supported by NASA and AFOSR.
†Associate Professor, Department of Aeronautics and Astronautics.

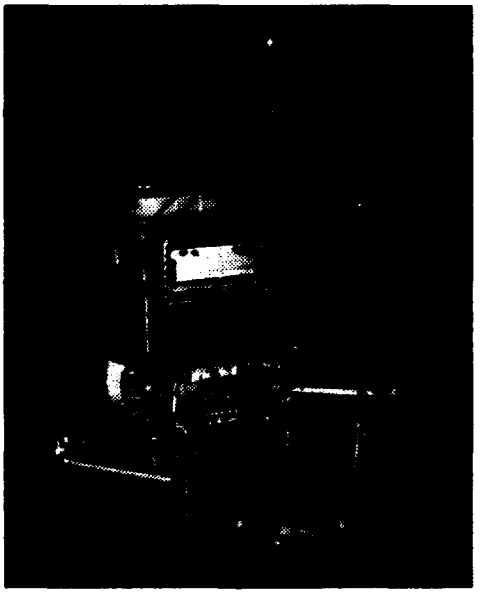
The main benefit is not that it maximizes the connections to neurons ratio, but instead that when combined with a systematic weight pruning procedure, it allows a more flexible use of layering. There has been a recent trend in using not one but two hidden layers; this is a generalization of the trend.

This general architecture makes full use of the backpropagation algorithm, while still allowing the use of modifications, such as the use of FIR connections in place of weights or backpropagation through time [3] [4]. Figure 1 shows the extra connections that are unused in a single-layered network. The question is whether the enhanced functionality outweighs the increased computational load and susceptibility to over-fitting. This must be decided for each application.

In this application, the extra connections are found to be useful when coupled with a procedure to control over-fitting. In particular, the 3x4 matrix in the upper right corner of the weight matrix provides direct linear information flow from input to output (sigmoids are used only for the outputs of hidden neurons), and the 3x3 upper triangular matrix in the lower right corner provides communication between outputs. While these functions could be provided with processing components in series or parallel with the network, the fully-connected architecture provides a seamless integration of these capabilities.

There are two objectives in this paper. First, the advantages of a fully-connected architecture are demonstrated by applying it to a specific thruster mapping problem characteristic of a space robot. A procedure for handling the over-fitting issue is also demonstrated in this context.

The second objective is to demonstrate experimentally that the advantages are realizable. This is done by performing experiments on a mobile robot which operates in two-dimensions, using air-cushion technology to simulate the drag-free and zero-g characteristics of space. This robot, shown in Figure 2, is a fully self-contained 2-D model of a free-flying space robot complete with on-board gas, thrusters, electrical power, multi-processor computer system, camera, wireless Ethernet data/communications link, and two cooperating manipulators. It exhibits nearly frictionless motion as it floats above a granite surface plate on a 50 micron thick cushion of air. The three degrees of freedom (x, y, θ) of the base are controlled using eight thrusters positioned around its perimeter [5].



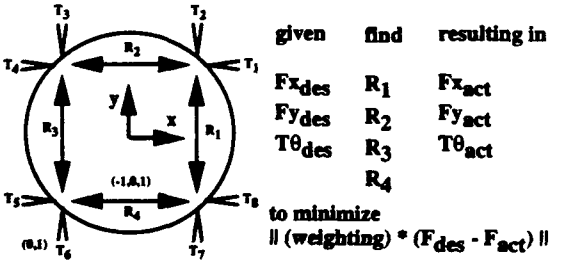


Figure 2: **Stanford Free-Flying Space Robot, Thruster Mapping Problem Definition**

An overview of the thruster mapping problem is presented first to provide a basis for describing the development and demonstration of the benefits of applying a fully-connected neural network.

2 Thruster Mapping Problem Definition

The overall objective is to use a set of eight full-on, full-off air jet thrusters to approximate a continuous-valued force vector that is commanded by the position/attitude control law of the mobile robot. The neural

network determines the combination of thrusters to fire that will generate a (normalized) resultant force vector as close as possible to that commanded. Note that the control design "ignored" the restrictions of on-off actuators, and assumed that feedback would compensate for errors due to on-off thrusters.

The right side of Figure 2 shows the eight (0,1) thruster locations, $[T_1, T_2, ..., T_8]$, as well as the equivalent set of four (-1,0,1) reaction forces, $[R_1, R_2, R_3, R_4]$. The thruster mapping task to be performed during each sample period is to take an input vector of continuous-valued desired forces, $[F_{xdes}, F_{ydes}, \tau_{\theta des}]$, and find the output vector of discrete-valued thruster values, $[T_1, T_2, ..., T_8]$, that minimizes some cost function.

In general, there are 256 possible thruster firing combinations ($2^8 = 256$) that need to be searched to find the optimum. Fortunately, elimination of redundant combinations, as well as the use of forward and inverse transformations to exploit symmetries in the system reduce the search space to 11 [5].

The cost function employed in evaluating the search was a weighted sum of squared errors in the forces achieved. The weighting factors were the effective mass/inertia through which each thrust acted. Consequently, the performance index attempted to match resulting accelerations rather than absolute force values. That is $J = F_{err}^T \cdot Q^2 \cdot F_{err}$ where J is the cost, $F_{err} = F_{des} - F_{act}$, and Q is a diagonal matrix with $[1/M \; 1/M \; r/I_\theta]$ on the diagonal. The thruster firing pattern that minimizes this cost defines the "optimal" mapping which the neural network will emulate.

3 Development of Network

Network training was performed using supervised learning with the backpropagation algorithm to emulate this "optimal" mapping. Batching, and adaptive momentum and learning rate were used to speed up learning. Although these and other modifications to standard backpropagation yielded significant increases in learning performance, they are not the focus of this paper and will not be discussed further.

Careful selection of a pseudo-random training set was found to be helpful in representing the whole problem within a concise training set. A larger, separate test set was formed to evaluate performance of the network, but was not used for training purposes.

3.1 Value of Linear Feed-Through Connections

One of the advantages associated with the fully-connected network is that it provides the ability to implement a linear mapping between the input and output directly. This is advantageous for two reasons. First, if the optimal mapping has a strong linear component, then providing a portion of the network where this can be implemented directly can improve the rate of learning.

Second, if a linear approximation to the solution is known *a priori*, this mapping can be precomputed and loaded into the network before training begins. In this way, the remainder of the network can be used only to implement the nonlinear correction to the *a priori* estimate. This will also increase the rate of learning. Note that the weights associated with the linear component can be frozen or included in the learning depending on the confidence associated with the *a priori* estimate.

For example, in the thruster mapping problem considered here, a reasonable linear approximation to the map can be found by assuming that the thrusters are capable of continuous-valued thrust output (a linearized version of this problem). The solution is simply a 4x3 pseudo-inverse of the 3x4 matrix which maps reaction forces, R, to base forces, F. Recognizing that the direct feed-through segment of the fully-connected network provides exactly this computation (output = weight matrix × input), it is possible to incorporated this *a priori* knowledge by putting the pseudo-inverse linear solution into that sub-matrix, as an initial condition for the weight matrix.

Figure 3 compares learning histories (thruster mapping error on the training set) for three cases in the thruster mapping example, each with 5 hidden neurons. Benefits of high initial learning rate and enhanced functional capabilities of the the fully-connected network can be seen. The fully-connected network initially learns very quickly, due to the weight gradient being instantly available due to the direct connection of inputs to outputs. For the 3-5-4 layered network, the hidden neurons must first become sufficiently activated to get a strong weight update signal, and it is easy for highly activated neurons to attenuate a backpropagating error signal. The fully-connected network with the *a priori* solution built in provides the best early performance.

After the startup phase, the layered performance surpasses that of the fully-connected, due to the reduced number of parameters, and simplified search space; however, while the layered network performance stops

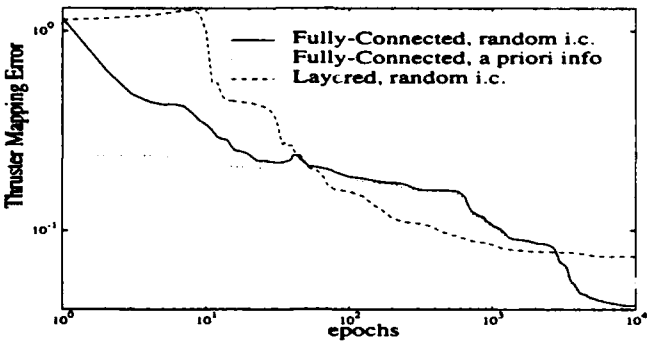


Figure 3: **Training History, Fully-Connected versus Layered**

improving after 1000 epochs, the fully-connected network eventually makes use of its greater functionality to surpass the performance of the layered network.

Observing the weight activation levels during training, the feed-through connections quickly converge, then the others gradually fall into place. By freezing the feed-through weights after an initial learning stage, the "moving target" problem is reduced. This refers to the weights changing directions and back-tracking throughout training while the network approaches a final solution. While this is not necessarily bad, it can slow down learning. The cascade-correlation algorithm takes this concept to its extreme–adapting weights for one neuron at a time, then freezing them before adding the next neuron [6].

3.2 Value of Input and Output Interconnections

The above advantages of the fully-connected network result from direct connections between the input and output layers. Advantages also result from connections between outputs (inputs). The weights associated with these are shown as the upper-triangular 3x3 (2x2) matrix in the lower right (upper left) corner in Figure 1. These weights allow one output to excite or inhibit a higher-numbered output.

The utility of this can be seen in the thruster mapping problem. If the network were to have an output (0,1) for each of the eight thrusters, and during training, a penalty were put on gas use, then the network could use this segment to allow the firing of one thruster to prohibit the firing of the opposite thruster (which would provide zero net thrust and waste fuel). This can be programmed manually prior to learning if observed. Note that the ability to implement this communication between output neurons is unavailable in a layered network.

Another example is the ability to avoid producing a bad result which is the average of two equally good results: if [1 1 -1 -1] and [-1 -1 1 1] both minimize the cost, they may be equally likely to activate when that force is requested, resulting in [0 0 0 0]. The output interconnections would allow the network to use the first output to send it to either of the acceptable solutions, and avoid the ambiguity.

Cross-talk between all outputs would be beneficial, but backpropagation is limited to uni-directional information flow. This may make it important to select carefully the ordering of inputs and outputs. If more complicated, non-linear communication is desired, extra neurons may be placed between individual output or input neurons.

3.3 Architecture Selection to Avoid Over-fitting

The above has shown the potential value of the extra connections associated with a fully-connected neural network, but an important issue that must be addressed is the rapid increase in network parameters with each additional hidden neuron. A single-layered network with i inputs, h hidden neurons, and o outputs has $(i+o)\cdot h$ weights, while a fully-connected network has $((i+h+o)\cdot(i+h+o-1)/2)$ weights, not counting bias weights. The high number of parameters, while increasing functionality, makes the network susceptible to over-fitting.

It is common that during training, performance on test and training sets will improve until a certain point is reached when the network is no longer finding a better fit to the general problem, but is simply

fitting the particular data set. Use of a "sufficiently-large" training set can reduce over-fitting problems, but this may not be practical due to a lack of data, or an adaptation speed requirement that needs a faster solution than this brute-force approach.

The task is to find out which connections are useful and build a network using those weights, and no more. The network architecture selection could be performed manually. For this problem, a network with feed-through connections, weights corresponding to a 3-5-4 layered network, and output interconnections, would probably work well, and not be susceptible to over-fitting, given a good training set.

This heuristic approach may overlook some valuable extra connections, and may still result in over-fitting, so some sort of systematic network pruning technique is required. One used successfully involves the addition of a complexity cost term to the total cost function, as first proposed by Rumelhart [7] [8]. Each weight contributes $\lambda \cdot (w_s^2/(w_s^2+1))$ to the total cost function, where $w_s = w/w_0$ is a scaled value of the weight. Scale factor, w_0, effectively sets the cutoff point for weights, and λ selects the relative importance of complexity cost versus performance cost. A similar cost function was used to eliminate entire neurons: if the collection of weights into and out of a certain neuron are small, those weights will be reduced.

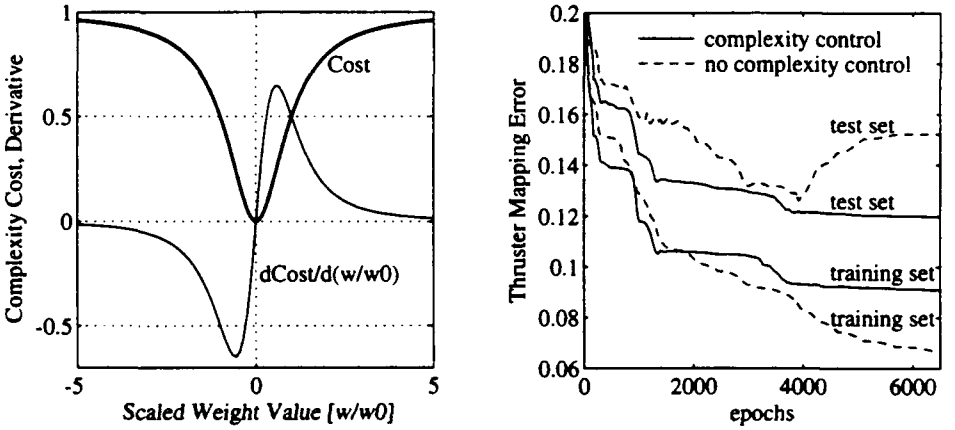


Figure 4: **Complexity Control Function, Effect on Network Performance**

The complexity control function is plotted in Figure 4. Very small weights cost little. Large weights (indicating that they contribute significantly to the network's function) cost λ, but the gradient is small, so there is little incentive to decrease them. Small weights (probably do not significantly help performance) must fight a strong gradient and are likely to be driven to zero. In contrast, a cost function like $\lambda \cdot (w_s^2)$ would be simpler, but it would reduce all the weights, whereas this one only reduces the relatively small weights, leaving the useful ones unrestrained.

Training histories for a fully-connected network with 5 hidden neurons (78 weights) are also shown in Figure 4. The training set has 80 patterns, and the test set 1000. Without any sort of complexity control, over-fitting becomes clear at around the 4000th epoch, as the performance on the test set worsens, while performance on the training set improves. With the addition of the complexity term, this problem is controlled, as performance histories on test and training sets no longer diverge.

4 Experimental Results

The robot gets position feedback from an overhead "global" vision system. A pair of overhead CCD cameras detect LED's on top of the robot, and an off-board vision processing board computes the robot position and velocity. This information and command signals are transmitted over a wireless Ethernet data/communications link to the vehicle controller on-board the robot.

The neural network thruster mapping component is implemented on the on-board Motorola® 68040 processor, as is the rest of the control system. Running at a 60 Hz sample rate, the calculations use only a small fraction of the available processing power.

Figure 5 shows robot base position during single-axis and multi-axis maneuvers. A simple neural network control component designed using backpropagation through time [2] [4] is used in conjunction with the neural

network component described in this paper. For the single-axis maneuver (left), good tracking is obtained from both optimal and neural network thruster mapping components. In the 3-axis maneuver (right), the neural network control system closely follows the 20-second-long straight-line trajectory in (x, y, θ). To avoid clutter, only the robot's actual (x, y) position is plotted.

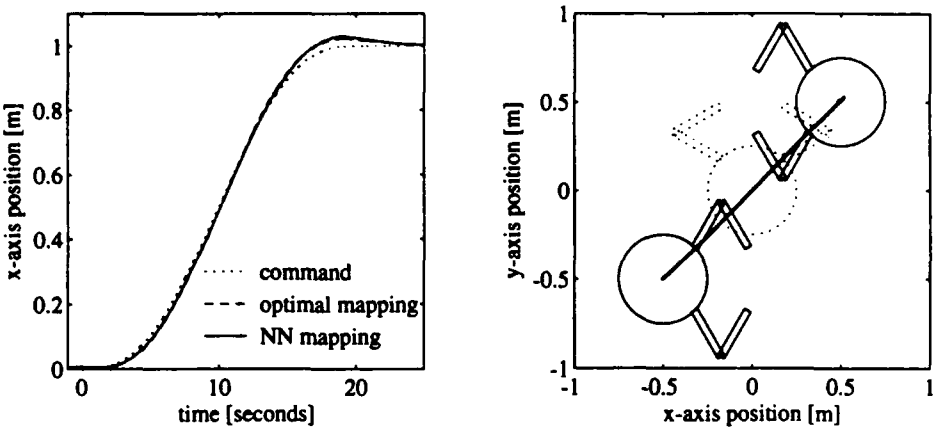


Figure 5: **Experimental Results, Single-Axis Maneuver, Multi-Axis Maneuver**

5 Summary and Conclusions

This paper has described initial steps in development of a neural network control system for a laboratory-based model of a free-flying space robot. A fully-connected neural network, aided by a systematic complexity control scheme, was trained with supervised learning to emulate a complex thruster mapping function typical of spacecraft. The increased connectivity afforded by the layer-less, fully-connected architecture was found to yield specific benefits in this application.

The neural network thruster mapping component was implemented on a 2-D laboratory model of a free-flying space robot. Combined with a simple neural network control component trained using backpropagation through time, it provided accurate tracking during multi-axis trajectories.

References

[1] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing*, page 318. The MIT Press, Cambridge, MA 02142, 1986.

[2] Paul J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, October 1990.

[3] E. Wan. Temporal backpropagation for FIR neural networks. In *International Joint Conference on Neural Networks*, pages 575–580, San Diego CA, June 1990.

[4] Derrick H. Nguyen and Bernard Widrow. Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, 10(3):18–23, April 1990.

[5] Marc Ullman. *Experiments in Autonomous Navigation and Control of a Multi-Manipulator Free-Flying Space Robot.* PhD thesis, Stanford University, Stanford, CA 94305, January 1993.

[6] Scott E. Fahlman and Christian Lebiere. The cascade-correlation learning architecture. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems 2.* Morgan Kaufmann Pulishers, 1990.

[7] David E. Rumelhart. Learning and generalization. In *Proceedings of the IEEE Int. Conf. Neural Networks*, San Diego CA, 1988. (plenary address).

[8] Andreas S. Weigend, Bernardo A. Huberman, and David E. Rumelhart. Predicting the future: A connectionist approach. *International Journal of Neural Systems*, 1(3):193–209, 1990.